Chapter 12
Including Students With Disabilities in the Coding Classroom

Tess Levinson
Tufts University, USA

Libby Hunt
Tufts University, USA

Ziva Hassenfeld
Brandeis University, USA

ABSTRACT

This chapter discusses understandings of coding and computational thinking education for students with disabilities. The chapter describes the special education system in the United States, including limitations in how computer science education is made available to students receiving special education services. The chapter then provides a summary of research in computer science education for students with disabilities, including both high-incidence and low-incidence disabilities. A case study of a young student with a mild disability learning in a general education computational thinking program is then presented, and the implications of the case study for future research directions are discussed.

DOI: 10.4018/978-1-7998-7308-2.ch012

INTRODUCTION

Sophie[1], a 5-year-old girl enters Ms. Locke's kindergarten class smiling. She sits on her spot on the carpet and waits for class to begin. The girls around her argue over who will sit next to who. She seems intentionally oblivious. At some point, she is drawn into this seating dance as another student asks if she will move so that said girl can sit next to another student. She obliges, undisturbed. Soon Ms. Locke begins class and sings the robot part song. Sophie stands delighted and dances along, moving side to side: "The body is connected to the motor; the motor is connected to the... so move robot move." Today is the day the teacher informs the class they will finally get to play with the KIBO robot. The teacher has organized and sorted all the KIBO parts into different storage bins in the "materials" part of her classroom. The students are broken into pairs and called into a line to collect their materials. Sophie and her partner, Pete, wait patiently as students mull over the KIBO bins. Finally, it is their turn. Sophie and Pete take turns filling their tray with all the KIBO parts.

Soon they find a quiet place on the rug and begin building. They work collaboratively, taking turns, co-constructing a path for the KIBO robot to travel and the corresponding program that will allow KIBO to travel. Sophie plans her project in her Design Journal and references that plan as she and Pete create their program. Instead of becoming discouraged when the scanning of the coding blocks does not work, they work together to problem-solve, and Sophie scans the coding blocks with the robot. One would not know from this short snapshot of the classroom that Sophie does not talk in school. She doesn't speak out loud to Pete as they build their program, and her design plan does not include the voice recorder and associated blocks. Still, she and Pete work together, excited by the possibilities KIBO offers for creativity and expression. In this chapter we will explore what Sophie's teacher did to accommodate her disability so that she could access KIBO learning alongside her peers. More broadly, we will discuss how computer science education can be used towards inclusive classrooms and pedagogy.

STUDENTS WITH DISABILITIES AND COMPUTER SCIENCE

Fourteen percent of public-school students in the United States ages 3-21 receive special education services under the Individuals with Disabilities Education Act (Congress, 1975) for some form of disability, which can range from specific learning disorder, to speech or language impairment, to autism spectrum disorder (*Students with Disabilities*, 2020). As each student's individual needs vary, so do the special education services provided. A student with a *high-incidence disability*, a category including but not limited to learning disabilities, emotional and/or behavioral disorders, and speech or language impairments, may spend most of their day with their peers in the general education classroom and only receive an hour or so of special education services for domain-specific instruction (Gage et al., 2012). Students with disabilities of this nature comprise the majority of students with disabilities (Gage et al., 2012; National Center for Education Statistics, 2021). In contrast, students with *low-incidence disabilities* have disabilities that affect learning across domains, such as significant sensory or cognitive impairments (Congress, 1975). Depending on the nature of their disability and needs, students with more-significant intellectual disabilities or other domain-general disabilities may spend most of the day receiving special education services, meaning much of their education is provided by the special education teacher. As suggested by the term, the minority of students with disabilities have disabilities that are classified as low-incidence.

Over 60% of students with disabilities spend more than 80% of their day in the general education classroom (National Center for Education Statistics, 2021). However, students with disabilities do not have equal access to computer science and computational thinking education as their nondisabled peers, which ultimately leads to knowledge gaps for students with disabilities in increasingly important 21st century skills. For example, while approximately 10% of students without disabilities scored below proficient for the National Assessment of Educational Progress technology and engineering literacy content area, nearly half of students with disabilities scored below proficient (National Center for Education Statistics, 2021). Groups and initiatives such as AccessCSForAll and Deaf Kids Code are increasing access to computer science and computational thinking programming for kids with disabilities (deafkidscode.org, n.d.; Ladner & Israel, 2016). Additionally, researchers are developing dedicated educational programs for students with disabilities, as well as best practices for accommodation, in order to improve the quality of computer science education for these students.

Many of the specific programs and interventions relating to computer science and coding instruction for students with disabilities have focused on developing educational programs for students with low-incidence disabilities and autism (Taylor, 2018). Much of this research focuses on educational pedagogies based around explicit instruction. In a curriculum based on explicit instruction, a student might learn, for example two control structures, and then practice them by programming a specific game. Using evidence-based explicit instruction, computer programming has been taught to students with Down syndrome, autism, and intellectual disability (Pivetti et al., 2020). For example, Knight, Wright, and DeFreese (2019) used an explicit instruction pedagogy to teach an elementary student with autism and significant behaviors to code using the Ozobot robot. Following the instruction period, the student was able to generalize the coding skills to new coding challenge (Knight et al., 2019). However, skills taught through explicit instruction do not necessarily generalize. This means a child may be able to use a skill within a specific setting but cannot use the skill in a new setting or to create an unknown program. For example, Taylor (2018) used explicit instruction to teach preschool, kindergarten, and first grade students with intellectual disabilities to use the Dash robot, and although all the students learned to code the robot, no student was able to generalize the skills to complete a novel coding challenge (Taylor, 2018).

These evidence-based explicit instructionist pedagogies used by special educators are in tension with the constructionist pedagogies for computational thinking (Bers, 2020). Constructionist models allow for student-driven play to drive learning, whereas explicit instruction provides a structure for learning. For example, Munoz et al (2018) taught students with autism to create video games using an instructionist pedagogy that provided students with the prompt, characters, and code (Munoz et al., 2018). Through this instructionist video-game learning curriculum, students with autism learned computational thinking skills such as abstraction, problem decomposition, and data representation. In contrast, in a constructionist robotics curriculum focused on cause and effect, students participated in guided free-play involving coding and sensors (Albo-Canals et al., 2018). The primary goal of Albo-Canals et al.'s (2018) research was understanding student engagement with educational robots, rather than computational thinking learning, but the findings suggest that the students gained some computational thinking knowledge, including sequencing and cause-effect. There has not yet been research specifically on computational thinking learning through constructionist curricula for students with disabilities.

Most research on computer science education for students with disabilities has focused on students with low-incidence disabilities and autism who may receive more significant accommodations or modifications to their educational materials. However, the majority of students with disabilities have high-incidence disabilities, and as mentioned above, most of them receive education at least partially within the general education setting (Gage et al., 2012; *Students with Disabilities*, 2020). Services for students with high-incidence disabilities, which include specific learning disabilities (e.g., reading disabilities, math disabilities), speech and language impairments, and emotional and behavioral disorders, are often targeted to a student's specific area of need. For example, a student with a specific learning disability in reading may receive special education services in literacy and language arts but might not receive individualized attention or accommodations in computer science. Bouck and Yadav (2020) showed that students with high-incidence disabilities in an upper elementary school resource room learned computational thinking concepts such as algorithms through a combination of explicit instruction and unplugged activities. They also suggest use of instructional methods such as pre-teaching vocabulary and providing information in multiple formats (Bouck & Yadav, 2020). Israel et al. (2015) reinforce the use of multiple instructional methods and emphasize the use of Universal Design for Learning practices, which uses multiple means of representation, action and expression, and engagement to create an inclusive and accessible curriculum (Israel et al., 2015).

Here, we describe a case study of a student with a disability served primarily in the general education classroom, selective mutism. Selective mutism is defined as "a complex childhood anxiety disorder characterized by a child's inability to speak and communicate effectively in select social settings, such as school" (American Speech-Language-Hearing Association, n.d.). The condition must cause impairment either academically or socially and must not be explained by another communication or developmental disorder (Viana et al., 2009). Although speech or language impairment is classified as high-incidence with regard to special education services, selective mutism is thought to be a relatively rare diagnosis, with prevalence estimated to be between 0.47% and 0.76% (Viana et al., 2009). There is no known single cause of selective mutism, and while there is evidence suggesting an association with anxiety disorders, some students also express externalizing behaviors or ADHD (Viana et al., 2009). The complexity and variations of the disorder create further challenges for a teacher of a student with selective mutism, as there is no single approach to accommodate a student with this diagnosis. The curriculum presented in this case study was not intended as a program or intervention to teach computer science or coding to students with disabilities. Rather, by accommodating the needs of a student with a disability, the teacher was able to create an inclusive and accessible constructionist, coding environment. As such, the case study we present explores exciting new possibilities for using constructionist pedagogies in teaching computer science with students with high-incidence disabilities.

CASE STUDY: CODING AS ACCESSIBLE COMMUNICATION

At first or even second glance, Sophie's classroom participation was similar to that of any other child in her kindergarten class. She sat amid her peers during carpet circle times, raised her hand during participatory questions, and turned her head to anyone who addressed her. Sophie has selective mutism and does not speak, but she was fully included in her class's computer science programming. Sophie and her kindergarten class took part in a larger research project investigating how religious and secular elementary schools used tangible robotics as an opportunity to foster character development (see Chapter 10 in this book). As a research team, we were interested in the different ways that kindergarten-age children would interact with one another in the context of robotics, and how their classroom environment would influence those interactions.

Ms. Locke's classroom was a place where Sophie's disability was accommodated and accepted. Ms. Locke explained Sophie's disability to the researcher's when explaining an accommodation made to the curriculum, and throughout the implementation of the KIBO tangible robotics curriculum, Ms. Locke made notes about how she modified discussion-based activities to allow non-verbal participation. Ms. Locke made turned open-ended questions into "raise your hand if you agree" questions, allowing her to contribute non-verbally without standing out among her peers. Ms. Locke also seemed to have an eye out for Sophie. In one classroom activity we observed, we watched as Sophie began to look a little despondent while her peers shouted their ideas. Ms. Locke noticed and turned to Sophie, saying, "Tell me, should we do a dog?" Sophie smiled and nodded. The acceptance and accommodation of Sophie's disability modeled by Ms. Locke appeared to translate to the other students' acceptance and inclusion of Sophie. In another activity, while creating underwater scenes with crayons, Sophie's classmate leaned over the table to look at her drawing. "I love yours! Look how Sophie did hers!" her classmate remarked, drawing everyone's attention to Sophie. "So pretty," another classmate said. Sophie did not look up but smiled slightly and continued coloring.

Throughout the tangible robotics curriculum, Sophie had the same partner, her classmate Pete. In her notes, Ms. Locke writes that Pete "continues to show kindness and patience towards his partner. Sophie is very quiet, and Pete takes time to explain/talk with Sophie about KIBO." In the hands-on robotics activity, Pete and Sophie worked to build the KIBO robotics kit together. Sophie poked Pete to get his attention. He never denied her the chance to touch the KIBO robot even when she was having difficulty scanning the tangible programming blocks. Ms. Locke wrote in her lesson notes: "Pete didn't take KIBO away and didn't do the scanning himself, he just held his friend's hands from above and controlled her hand movements." At another point in the curriculum, Ms. Locke used Sophie and Pete's program as an example for the whole class. Although Pete and Ms. Locke did the verbal presentation, they consistently used the plural pronouns "them and their" to give ownership to Pete and Sophie, not just Pete. In another class discussion about who helped other students work with their KIBOs, Pete raised his hand. Sophie noticed and raised her hand. Ms. Locke called on Pete. He announced to the class that Sophie had helped him because she scanned the barcodes of the tangible block program for him. Sophie smiled big and looked down, but the smile lingered for moments after.

We found through analysis of our ethnographic data and video observations that Sophie demonstrated more communicative acts during KIBO robotics activities than during discussion-based activities. She ran from spot to spot with her classmates during "Robot Corners," a game about differentiating between items that are or are not robots, but during sharing circles, she appeared distracted and uninterested. While this finding may feel intuitive, this serves as a reminder of the role of tactile and kinesthetic learning tools for students with communication-related disabilities. The fact that this particular robotics kit, KIBO, centers on student expression and the teaching of coding as a language for communication, makes this finding even more promising for future applications of KIBO as a tool for students with communication-related disabilities to learn computational thinking.

While Sophie offered consistent communicative gestures whenever she was engaged with the KIBO robotics kit, no activity in the curriculum showed her engagement with the tangible tool more than her final project. Figure 1 below shows her planning sheet in her Design Journal for her final project. The assignment asked the students to create "Gratitude Floats" celebrating things special to the students and their community: Because this lesson took place close to Thanksgiving, this was an opportunity for the students to examine the tenets of their school and reflect on what they were grateful for.

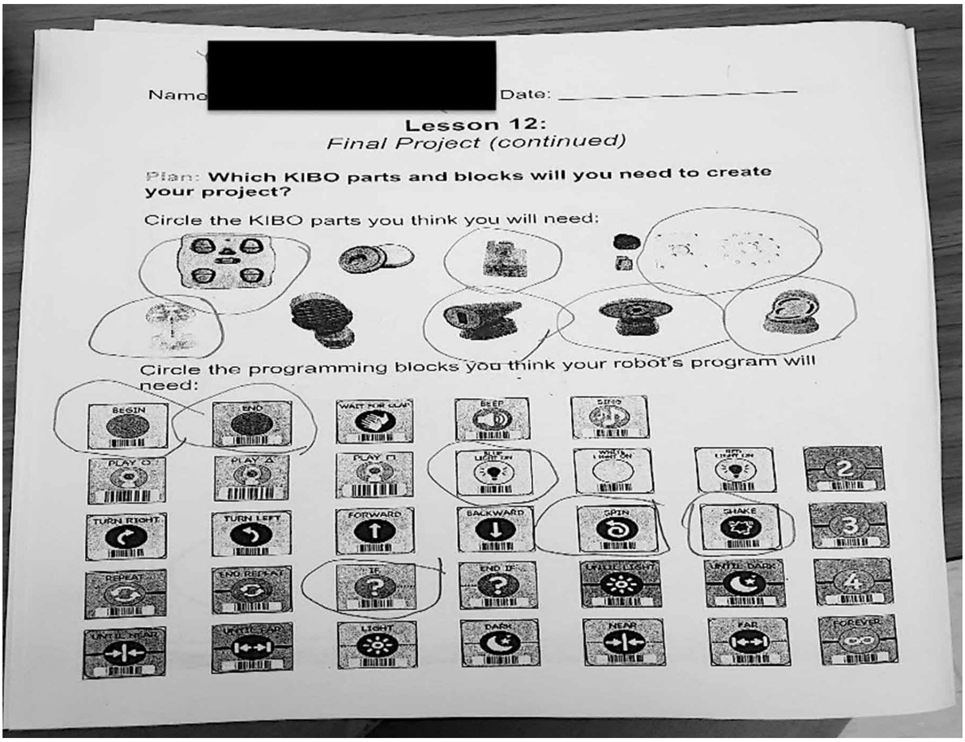
GRATITUDE FLOATS (15 min) Ask students to think about what makes their school special. Often, things that are special to you have some sort of meaning that signifies who you are or where you come from. Tell students that today, they will be making "Gratitude Floats," similar to a Thanksgiving Parade, celebrating their school and what makes it special. Ask students: What's important to you? Is it important to other people in the school too? What is different about our school than other schools? Students then should draw images of the things they felt made their school special. These images will later be used to decorate their Gratitude Floats.

The project continued:

PLAN THE PARADE (15 min) Before giving the students their KIBO, have the students plan out their parade. They should imagine if they could take their parade around the school, where they will go (e.g., other classrooms, the cafeteria, the chapel) and why. If time allows, children could even draw their route in the form of a map in their Curiosity Journals.

The blocks Sophie circled in her project plan suggest that she had a developing technical understanding of the KIBO programming language. First, she circled that she would use both a begin block and end block, both necessary for any KIBO program. This is significant because it ties into a basic understanding of the foundations of programming and connects to the powerful idea of algorithms and sequencing. Second, she circled movement blocks in her program, suggesting an expanded vocabulary of programming functions. Third, she circled both the light bulb and the light block, suggesting an emergent understanding of hardware-software correspondence. Although she did not yet show a mastery of this concept, for example, selecting the light and distance sensors without the corresponding blocks, this is significant as it connects to the powerful ideas of representation and multiple tools of communication.

Figure 1. Sophie's final project plan (IGI, 2021)
Source: IGI, 2021



While Sophie's project planning sheet showed her technical understanding of the KIBO robotics kit and block programming language, the sheet also revealed that Sophie saw the KIBO programming language as a language that she could use, express in, and communicate with. Particularly noteworthy was that Sophie felt empowered to circle every sensor except the voice recorder. Ms. Locke created a classroom culture in which Sophie was included and her disability was accommodated, and Sophie was comfortable in this classroom to express herself using every accommodating aspect of the KIBO language while rejecting the unaccommodating aspects. Within the KIBO language, she was able to advocate for and accommodate her own needs, making the language work for her.

CONCLUSION

Sophie's classroom experience suggests that even young children with disabilities can access mainstream, constructionist computer science learning environments with classroom accommodations, and that this opportunity to explore the coding platform leads to creative expression and classroom communication using the coding language. Her planning sheet communicates that she felt empowered to use the KIBO robotics kit to build her "Gratitude Float." She communicated in her plan that the program would require use of all the robotic sensors, except one, the voice recorder. With the classroom accommodations provided by her teacher, she could compose and self-express using the KIBO robotics language and was able to write programs expressing and accommodating her individual needs.

Sophie's successful experience reinforces previous research on students with disabilities and computer science on how to incorporate Universal Design for Learning and other accommodations into computer science instruction (Israel et al., 2015). For example, Israel et al. suggest that teachers give students with disabilities roles within project groups that allow them to focus on their strengths, while altering expectations for the student as necessary (Israel et al., 2015). While working with Pete, Sophie scanned the code (a non-verbal task), while Pete verbally shared their work with the class. The constructionist tangible robotics curriculum used in Ms. Locke's class also used many of Israel et. al.'s suggested practices, for example by providing the students with a culturally-relevant project or including unplugged activities to provide for multiple means of action and expression (Israel et al., 2015). Sophie's success with this curriculum suggests that teachers can use these practices to create an inclusive and accommodating coding classroom even for students as young as Kindergarten.

Computer science education for students with disabilities is important. These students are entitled to equally access all educational opportunities as their non-disabled peers, including computer science education (IDEA, 2004). Although most students with disabilities do not have computer-science or robotics specific accommodations, previous research suggests applying the supports already in place for other classroom subjects will lead to successful learning outcomes in computer science for students with disabilities (Snodgrass et al., 2016). We saw this with Sophie, who was included, engaged, and ultimately successful in the student-centered tangible robotics curriculum because her teacher's existing supports allowed for alternate methods of communication. For other students, existing supports might include access to assistive technology, KIBO blocks modified to include braille, or multiple modes of providing instructions.

Recently, there have been increasing opportunities for students with disabilities to learn computer science and access computer science curricula. As mentioned earlier, organizations and initiatives such as AccessCSForAll and Deaf Kids Code are bringing computer science opportunities to more students with disabilities (deafkidscode.org, n.d.; Ladner & Israel, 2016). Educational programs in robotics and computational thinking are being developed and assessed for students with disabilities using traditional special education practices (Knight et al., 2019; Munoz et al., 2018; Taylor et al., 2017). As computer science education becomes more available to young children, students with disabilities have the right to learn these 21st century skills alongside their nondisabled peers. Our work with Sophie suggests even young students with disabilities can learn computer science in student-centered learning environments alongside their nondisabled peers, including experiencing the benefits of the student-centered computer science pedagogy. By expanding their existing supports to new computer science curricula, teachers can offer inclusive and exciting computer science opportunities to engage students with and without disabilities in new ways of thinking and expression.

ACKNOWLEDGMENT

This research was supported by the Templeton World Charity Foundation.

REFERENCES

Albo-Canals, J., Martelo, A. B., Relkin, E., Hannon, D., Heerink, M., Heinemann, M., Leidl, K., & Bers, M. U. (2018). A Pilot Study of the KIBO Robot in Children with Severe ASD. *International Journal of Social Robotics*, *10*(3), 371–383. doi:10.100712369-018-0479-2

American Speech-Language-Hearing Association. (n.d.). *Practice Portal: Clinical Topics: Selective Mutism.* American Speech-Language-Hearing Association. Retrieved February 15, 2020, from https://www.asha.org/Practice-Portal/Clinical-Topics/Selective-Mutism/#collapse_8

Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge. doi:10.4324/9781003022602

Bouck, E. C., & Yadav, A. (2020). Providing Access and Opportunity for Computational Thinking and Computer Science to Support Mathematics for Students With Disabilities. *Journal of Special Education Technology*. Advance online publication. doi:10.1177/0162643420978564

Congress, U. (1975). The Individuals with Disabilities Education Act–IDEA.

deafkidscode.org. (n.d.). *Our Story*. Retrieved February 15, 2020, from https://www.deafkidscode.org/our-story

Gage, N. A., Lierheimer, K. S., & Goran, L. G. (2012). Characteristics of Students With High-Incidence Disabilities Broadly Defined. *Journal of Disability Policy Studies*, *23*(3), 168–178. doi:10.1177/1044207311425385

Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., & Tapia, T. (2015). Empowering K–12 Students With Disabilities to Learn Computational Thinking and Computer Programming. *Teaching Exceptional Children*, *48*(1), 45–53. doi:10.1177/0040059915594790

Knight, V. F., Wright, J., & DeFreese, A. (2019). Teaching Robotics Coding to a Student with ASD and Severe Problem Behavior. *Journal of Autism and Developmental Disorders*, *49*(6), 2632–2636. doi:10.100710803-019-03888-3 PMID:30734176

Ladner, R. E., & Israel, M. (2016). For all" in" computer science for all. *Communications of the ACM*, *59*(9), 26–28. doi:10.1145/2971329

Munoz, R., Villarroel, R., Barcelos, T. S., Riquelme, F., Quezada, A., & Bustos-Valenzuela, P. (2018). Developing Computational Thinking Skills in Adolescents With Autism Spectrum Disorder Through Digital Game Programming. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 63880–63889. doi:10.1109/ACCESS.2018.2877417

National Center for Education Statistics. (2021). *Digest of Education Statistics: 2019*. U.S. Department of Education. https://nces.ed.gov/programs/digest/d19/

Pivetti, M., Di Battista, S., Agatolio, F., Simaku, B., Moro, M., & Menegatti, E. (2020). Educational Robotics for children with neurodevelopmental disorders: A systematic review. *Heliyon*, *6*(10), e05160. doi:10.1016/j.heliyon.2020.e05160 PMID:33072917

Snodgrass, M. R., Israel, M., & Reese, G. C. (2016). Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. *Computers & Education*, *100*, 1–17. doi:10.1016/j.compedu.2016.04.011

Taylor, M. S. (2018). Computer Programming With Pre-K Through First-Grade Students With Intellectual Disabilities. *The Journal of Special Education*, *52*(2), 78–88. doi:10.1177/0022466918761120

Taylor, M. S., Vasquez, E., & Donehower, C. (2017). Computer Programming with Early Elementary Students with Down Syndrome. *Journal of Special Education Technology*, *32*(3), 149–159. doi:10.1177/0162643417704439

The Condition of Education: Students with Disabilities. (2020). National Center of Education Statistics. https://nces.ed.gov/programs/coe/indicator_cgg.asp

Viana, A. G., Beidel, D. C., & Rabian, B. (2009). Selective mutism: A review and integration of the last 15 years. *Clinical Psychology Review*, *29*(1), 57–67. doi:10.1016/j.cpr.2008.09.009 PMID:18986742

ADDITIONAL READING

Bargagna, S., Castro, E., Cecchi, F., Cioni, G., Dario, P., Dell'Omo, M., Di Lieto, M. C., Inguaggiato, E., Martinelli, A., Pecini, C., & Sgandurra, G. (2019). Educational Robotics in Down Syndrome: A Feasibility Study. *Technology*. *Knowledge and Learning*, *24*(2), 315–323. doi:10.100710758-018-9366-z

González-González, C. S., Herrera-González, E., Moreno-Ruiz, L., Reyes-Alonso, N., Hernández-Morales, S., Guzmán-Franco, M. D., & Infante-Moro, A. (2019). Computational Thinking and Down Syndrome: An Exploratory Study Using the KIBO Robot. *Informatics (MDPI)*, *6*(2), 25. doi:10.3390/informatics6020025

Israel, M., Ray, M. J., Maa, W. C., Jeong, G. K., Lee, C. E., & Lash, T. (2018). School-Embedded and District-Wide Coaching in K-8 Computer Science: Implications for Including Students with Disabilities. *Journal of Technology and Teacher Education*, *26*(3), 471–501.

Israel, M., Jeong, G., Ray, M., & Lash, T. (2020). Teaching Elementary Computer Science through Universal Design for Learning. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 1220–1226. 10.1145/3328778.3366823

Knight, V. F., Wright, J., Wilson, K., & Hooper, A. (2019). Teaching Digital, Block-Based Coding of Robots to High School Students with Autism Spectrum Disorder and Challenging Behavior. *Journal of Autism and Developmental Disorders*, *49*(8), 3113–3126. doi:10.100710803-019-04033-w PMID:31055684

Ladner, R. E., Stefik, A., Naumann, J., & Peach, E. (2020). Computer Science Principles for Teachers of Deaf Students. *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 1–4.

Stefik, A., Ladner, R. E., Allee, W., & Mealin, S. (2019). Computer Science Principles for Teachers of Blind and Visually Impaired Students. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 766–772. 10.1145/3287324.3287453

Wille, S., Century, J., & Pike, M. (2017). Exploratory Research to Expand Opportunities in Computer Science for Students with Learning Differences. *Computing in Science & Engineering*, *19*(3), 40–50. doi:10.1109/MCSE.2017.43

KEY TERMS AND DEFINITIONS

Constructionism: A student-directed pedagogy in which students' learning is self-directed based on individual questions and interests.

Explicit Instruction: A structured, teacher-directed pedagogy in which teachers provide direct instruction to students, provide students with a scaffolded learning environment, and assess student learning based on correctness of answers.

General Education Environment: The learning environment (including curriculum, teachers, standards, social environment, and physical environment) provided to children without disabilities.

High-Incidence Disability: A category of disabilities that includes specific learning disorders, speech or language impairments, ADHD, and emotional and behavioral disabilities.

Individuals With Disabilities in Education Act: The law that mandates special education services be provided to students with disabilities, and that students with disabilities are entitled to a free appropriate public education in the least restrictive learning environment.

Low-Incidence Disability: A category of disabilities that affect learning across domains, such as significant sensory impairments or significant cognitive impairments.

Special Education Services: Services provided by the school or school district to support students with disabilities, including special education teachers, paraprofessionals, and specialized curricula.

ENDNOTE

[1] All names are pseudonyms.